MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. No. 141 September 1967.

EUTERPE-LISP: A LISP System with Music Output.

by

Stephen Smoliar.

EUTERPE (A.I. Memo. No. 129), was designed as a "real-time music program" which would interpret music described as "voice-programs" in DDT. These voice-programs consisted of note words, description of tones to be sounded, and control words which determined the parameters of pitch, tempo, articulation and wave form and allowed for a subroutine feature and transfer within the voice-program. It had been hoped that complex musical forms could be described in terms of a few collections of note words and sequences of control words.

However, musical variation and development is more subtle than the developmental power of these control words. Any transformation of musical materials may be expressed as a LISP function; therefore, the control words were abandoned and EUTERPE was linked to LISP. The voice-programs would be written and loaded by LISP and played by EUTERPE. The principle function in the system is LOAD which takes two arguments: 1) an absolute location in core and 2) a list of note words. The note words are translated into EUTERPE-readable code and loaded into the proper voice program. The addresses of the first location of each of the six voice programs are SETQed by the system with the names VOICE1, ..., VOICE6. The value of LOAD is the next file word in core, so a series of lists may be loaded by the following bootstrapping procedure:

```
(SETQ LOC (LOAD VOICE1 LIST1))
(SETQ LOC (LOAD LOC LIST2))
. . .
```

An example is given at the end of this Memo.

A note is expressed as a dotted pair; the CAR denotes pitch, the CDR denotes duration. If the CDR is NIL, the duration is assumed to be the same as the preceding note (this is the same convention as EUTERPE; see example). The symbols are the same as in EUTERPE, but they are EXPLODED into lists. Hence middle C is now

```
(K C)
```

and a triplet-sixteenth note is

```
(16 T 3).
```

There are also a few "control-lists" which function as their analogs in EUTERPE; these are

(CANCEL n) (CDR may be NIL)
(START n)
(RELTEM n)
(TEMPO n)
(ARTIC n)
(WAVE n)
(FINE)

Finally, there are functions written into the system which may be used in describing music

---

MA — Major scale starting on pitch given as argument (ascending).
Example: (MA (QUOTE (K B))) has the value
((K B) (L C S) (L D S) (L E) (L F S) (L G S) (L A S) (L B))

NA — Natural minor scale (ascending).

HA — Harmonic minor scale (ascending).

ME — Melodic minor scale (ascending).

NUP — (NUP N I) is the pitch I half-steps above pitch N.

NDOWN — I half-steps below pitch N.

TRANSTONAL — (TRANSTONAL X K N) transposes a list of pitches, X, to begin on a new note N, in tonality K = ({MAJOR / MINOR} pitch)

TRANSPOSE — (TRANSPOSE X N) is a rigid transposition.

TR — (TR X N) takes a full list of input for EUTERPE and raises all pitches N half-steps.

INVERTONAL — (INVERTONAL X K) inversion of list of pitches, X, with respect to tonality K.

INVERT — (INVERT X) is rigid inversion.

RETROG — (RETROG X) is retrogression.

ROTN — (ROTN X N) rotates a list of pitches N steps.

Example: (ROTN (QUOTE ((J E) (J D) (J C)) 1)
has the value: ((J D) (J C) (J E)).

CHORD	argument is a list of up to six pitches; sounds as a chord until next input to LISP.
SETA	sounds A; alters tuning constant by numbers typed in, terminated by typing non-number.
TRANSFER	transfers to absolute location; (TRANSFER SETUP) prepares compilation; (TRANSFER PLAY) plays compiled version.

The user may also prepare his own function.

SETA is used as follows: The user types in the s-expression (SETA) and the machine sounds the pitch it assumes to be 440 cps. If this note is flat, the user types in a number (as an atom) the machine adds this number to its tuning constant, and it sounds a new pitch. If the note is sharp, the user types in a negative number. Once the machine is "in tune," any non-number (e.g., the atom OK) will terminate SETA and return to LISP.

The following program describes the Canon from Bach's Kunst der Fuge attached at the end of this Memo. The major portion of the program consists of SETQing the necessary thematic elements which are then loaded by the two PROGs at the end.

```
(SETQ IBASE 10.)
(SETQ T1 (QUOTE (
(ARTIC LEGATO)
((J D) . (8 T 3))
((J E))
((J D))
((J C S))
((J D))
((J E))
((J F))
((J G))
((J F))
((J E))
((J F))
((J G))
(ARTIC SLUR)
((J A) . (2 T))
(ARTIC LEGATO)
((J A) . (8 T))
(ARTIC SLUR)
((J G) . (8 T))
((J F))
(ARTIC LEGATO)
((J E) . (8 T))
(ARTIC SLUR)
((J F) . (2 T))
(ARTIC LEGATO)
((J F) . (8 T))
(ARTIC SLUR)
((J E) . (8 T))
((J D))
(ARTIC LEGATO)
((J C S) . (8 T))
((J D) . (4 T))
(ARTIC SLUR)
((J E) . (8 T))
((J F))
((J G))
(ARTIC LEGATO)
((J A) . (8 T))
((J G F) . (4 T))
(ARTIC SLUR)
((J G F) . (4 T))
(ARTIC LEGATO)
((J C S) . (4 T))
(R . (8 T))
((I A))
((I G))
((J C S))
(ARTIC SLUR)
((J D) . (8 T))
((J C S))
((I G))
(ARTIC LEGATO)
((I A) . (8 T))
(ARTIC SLUR)
((J E) . (8 T))
((J C S))
((I G))
(ARTIC LEGATO)
```

```
((I A) . (8 T))
(ARTIC SLUR)
((J F) . (8 T))
((J E))
((J D) . (4 T))
(R . (8 T))
((J B F))
((J A))
(ARTIC LEGATO)
((J G S) . (8 T))
(ARTIC SLUR)
((J A) . (8 T))
((J F))
((J E))
(ARTIC LEGATO)
((J D) . (8 T))
(ARTIC SLUR)
((J F) . (8 T))
((J E))
((J D))
(ARTIC LEGATO)
((J C S) . (8 T))
)))
(SETQ T2 (QUOTE (
(ARTIC LEGATO)
((J D) . (4 T))
(R)
((I A) . (8 T 3))
(PERT ((I B)) (1 . D))
((I A))
(PERT ((I G S)) (1 . D) (3 . D))
((I A))
(PERT ((I B)) (1 . D))
(PERT ((J C) . (8 T)) (2 . U))
((I A))
(PERT ((I B)) (1 . D))
(PERT ((J C)) (2 . U))
((J D))
((J E))
(PERT ((J F S)) (2 . D))
(PERT ((J G S)) (1 . D) (2 . D))
((J A))
((J B))
(PERT ((J F)) (1 . U))
((J E))
((J D) . (4 D))
(PERT ((J E) . (8 T)) (1 . D))
)))
(SETQ T3 (QUOTE (
(ARTIC LEGATO)
((J F) . (8 T))
(PERT ((J E)) (1 . D))
((J D))
((J C))
(PERT ((I B)) (1 . D) (2 . D))
((J C))
((J D))
(PERT ((I B)) (1 . D) (2 . D))
(PERT ((I G S)) (1 . D) (2 . D))
((I A))
```

```
(PERT ((I B)) (1 . D) (2 . D))
(PERT ((I E)) (1 . D))
(PERT ((J E)) (1 . D))
(PERT ((J C)) (2 . U))
((J D))
(PERT ((J E)) (1 . D))
(PERT ((I F S)) (1 . D) (2 . D))
((I A))
(COM (ARTIC STACD) (2 . (LEGATO)))
((J D) . (4 T))
(COM (ARTIC LEGATO) (2 . (LEGATO)))
(PERT ((I G S) . (8 T)) (1 . D) (2 . D))
(PERT ((I B)) (1 . D) (2 . D))
(COM (ARTIC STACD) (2 . (LEGATO)))
((J E) . (4 T))
(COM (ARTIC LEGATO) (2 . (SLUR)))
((I A) . (8 T))
(PERT ((J C)) (2 . U) (3 . D))
((J D))
(ARTIC LEGATO)
((J E) . (8 T))
(COM (ARTIC LEGATO) (2 . (SLUR)))
((J F) . (8 T))
((J D))
(PERT ((J C)) (2 . U) (3 . D))
(ARTIC LEGATO)
(PERT ((I B) . (8 T)) (1 . D))
(PERT ((J C) . (4 T)) (2 . U) (3 . D))
((J F))
((J D))
((J E))
)))
(SETQ T4 (QUOTE (
(ARTIC LEGATO)
((J F) . (8 T 3))
((J G))
((J F))
((J E))
((J F))
((J G))
((J A) . (4 D))
((J G) . (15 T))
((J F))
((J E) . (8 T))
(PERT ((J C)) (2 . U) (3 . D))
((J D))
((J E))
((J F) . (4 D))
((J E) . (15 T))
((J D))
((J D S) . (4 T))
((I A))
(R . (8 T))
((J D))
((J C))
((I B F))
((I A) . (4 T))
((I F) . (8 T))
((I A))
((J D) . (4 T))
```

```
(PERT ((I F)) (1 . U) (2 . J) (3 . D))
((I 3))
((I D))
(R . (8 T))
((I E))
(PERT ((I F)) (1 . U))
((I 3))
((I A))
((I 3))
(PERT ((I F)) (1 . U))
((I A))
((I B))
((I A))
((I 3))
((I B))
((J C))
((I B))
((I A))
((I 3))
((I A))
((J F))
(PERT ((J E)) (1 . D) (3 . D))
((J D))
(PERT ((J E)) (1 . D) (3 . D))
((J C))
((I A))
((J C))
(PERT ((I F S)) (2 . D) (3 . J))
((I A))
(PERT ((I G S)) (1 . D) (2 . D))
(PERT ((I B)) (1 . D) (2 . D))
((I A) . (4 T))
(R)
((J E) . (8 T 3))
((J F))
((J E))
((J D))
((J E))
((J F))
)))
(SETQ CS1 (QUOTE (
(ARTIC SLUR)
((J 3) . (4 T))
(ARTIC LEGATO)
((J 3) . (16 T))
((J F))
((J E))
((J D))
((J C) . (8 T 3))
((J D))
((J C))
((I B))
((J C))
((J D))
((J E) . (8 T))
((J D))
(PERT ((J C)) (1 . U))
((I B))
((I A) . (4 D))
((I B) . (8 T))
```

```
((J C))
((I B F))
((I A))
((I G))
((I F))
((I G))
((I A))
((I F))
((I B F))
((I G))
((I F))
((I E))
((I D) . (4 T))
(R)
(R)
((I A))
((I D) . (8 T))
((I F))
((I B F) . (4 T))
((I E) . (8 T))
((I G))
((J C) . (4 T))
(R . (8 T))
((I A))
((I G))
((I F S))
((I G) . (4 T))
((I A) . (2 T))
((I B) . (4 T))
((I C S) . (8 T))
((I D))
((I E))
((I F))
((I G) . (8 T 3))
((I A))
((I G))
((I F))
((I G))
((I A))
((I B F) . (4 T))
((I G))
((J D) . (8 T 3))
((J E))
((J D))
((J C S))
((J D))
((J E))
)))
(SETQ CS2 (QUOTE (
((L F) . (4 T))
((L D))
((K A) . (8 T 3))
((K B))
((K A))
((K G S))
((K A))
((K B))
((L C S))
((L D))
((L C S))
```

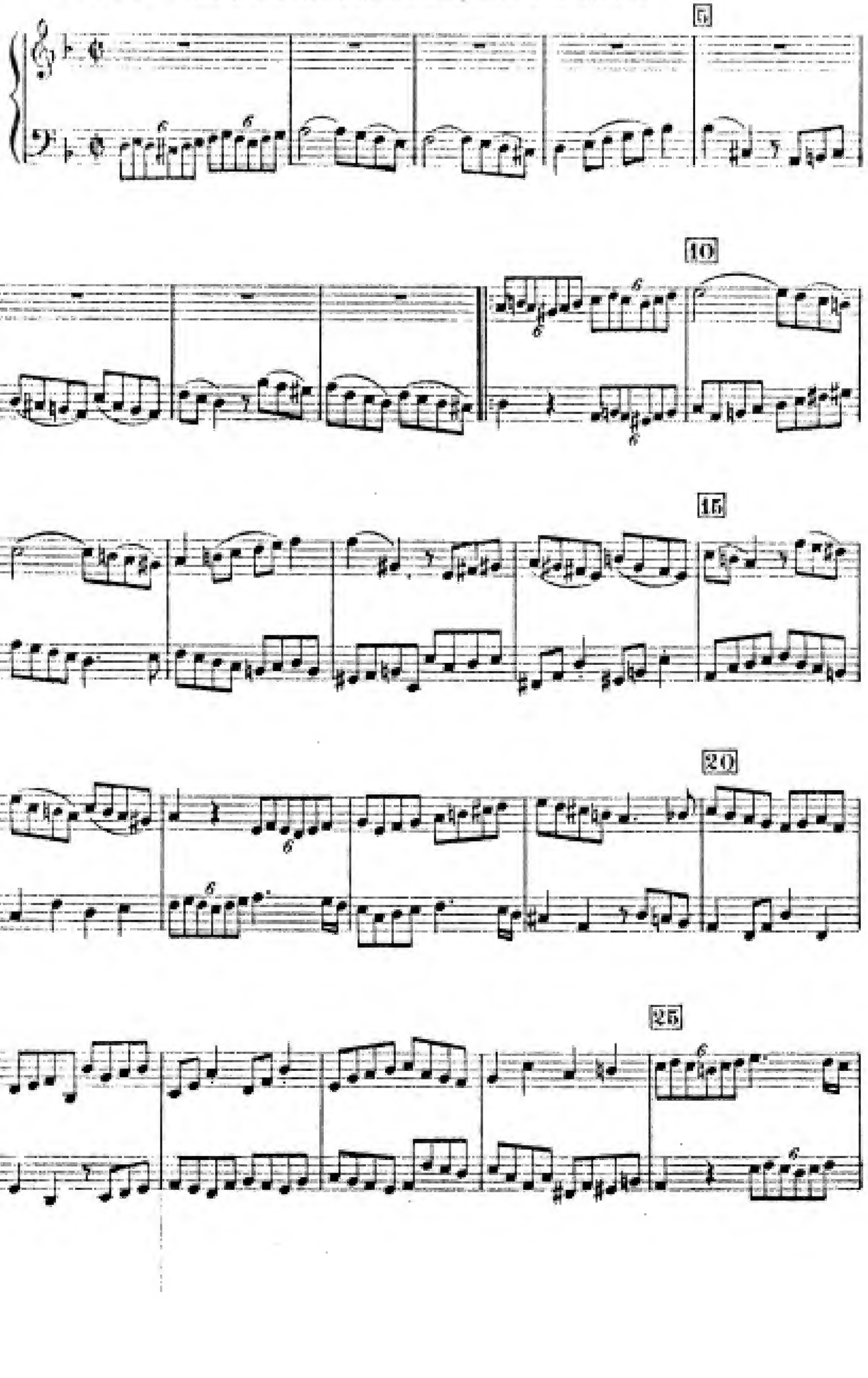
```
((K 3))
((L C S))
((L D))
((L E) . (8 T))
((K 4))
(ARTIC SLUR)
((L A) . (4 T))
(ARTIC LEGATO)
((L A) . (8 T))
((L 3))
((L F))
((L E))
(ARTIC SLUR)
((L F) . (2 T))
)))
(SETQ CS3 (QUOTE (
((J 3) . (8 T 3))
((J 4))
((J 3))
((J F))
((J 3))
((J 4))
((J 3 F) . (4 T))
((J 3))
)))
(SETQ CS4 (QUOTE (
((L F) . (8 T))
((L E))
((L D) . (4 T))
(R . (8 T))
((L 3 F))
((L 4))
((L 3 S))
(ARTIC SLUR)
((L A) . (8 T))
((L F))
((L E))
(ARTIC LEGATO)
((L D) . (8 T))
(ARTIC SLUR)
((L F) . (8 T))
((L E))
((L D))
(ARTIC LEGATO)
((L C S) . (8 T))
((L D) . (1 T))
(FINE)
)))
(SETQ CS5 (QUOTE (
((J D) . (8 T))
((I F))
((I 3))
((I 4))
((I 3 F))
((I 3))
((I F))
((I E))
((I F) . (4 T))
((I 3 F))
((I 3))
```

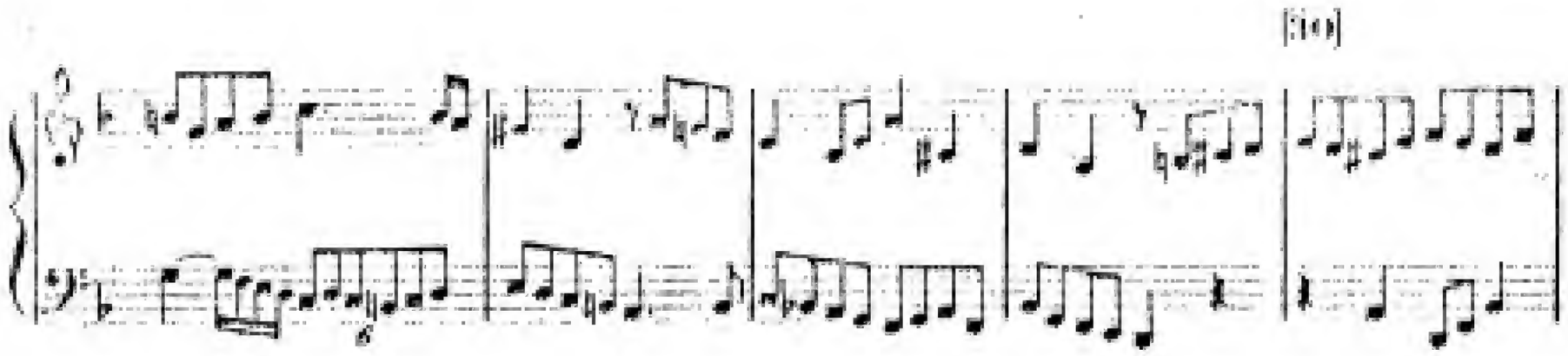
```
((I A))
((I D) . (1 T))
(FINE)
)))
(SETQ CS6 (QUOTE (
((J F) . (8 T 3))
((J G))
((J F))
((J E))
((J F))
((J G))
((J A) . (8 T))
((J D))
(ARTIC SLUR)
((K D) . (4 T))
(ARTIC LEGATO)
((K D) . (8 T))
((K C))
((J 8 F))
((J 4))
(ARTIC SLUR)
((J 8 F) . (2 T))
)))
(SETQ IBASE 8.)
(SETQ T2A (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (CADR X))
 (T X))))
T2))
(SETQ T2B (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 1 X)) (CADR X))
((EQ (CDR (ASSOC 1 X)) (QUOTE U)) (CONS (ENNOTE (SEMIUP (CAADR X))) (CDADR
X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
 (T X))))
T2))
(SETQ T2B (TR T2B 23))
(SETQ T2C (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 2 X)) (CADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (CONS (ENNOTE (SEMIUP (CAADR X))) (CDADR
X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
 (T X))))
T2))
(SETQ T2C (TR T2C 30))
(SETQ T2D (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 3 X)) (CADR X))
((EQ (CDR (ASSOC 3 X)) (QUOTE U)) (CONS (ENNOTE (SEMIUP (CAADR X)))
(CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
(T X))))
T2))
(SETQ T3A (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((OR (EQ (CAR X) (QUOTE PERT)) (EQ (CAR X) (QUOTE COM))) (CADR X))
 (T X))))
T3))
(SETQ T3B (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((OR (EQ (CAR X) (QUOTE PERT)) (EQ (CAR X) (QUOTE COM))) (COND
```

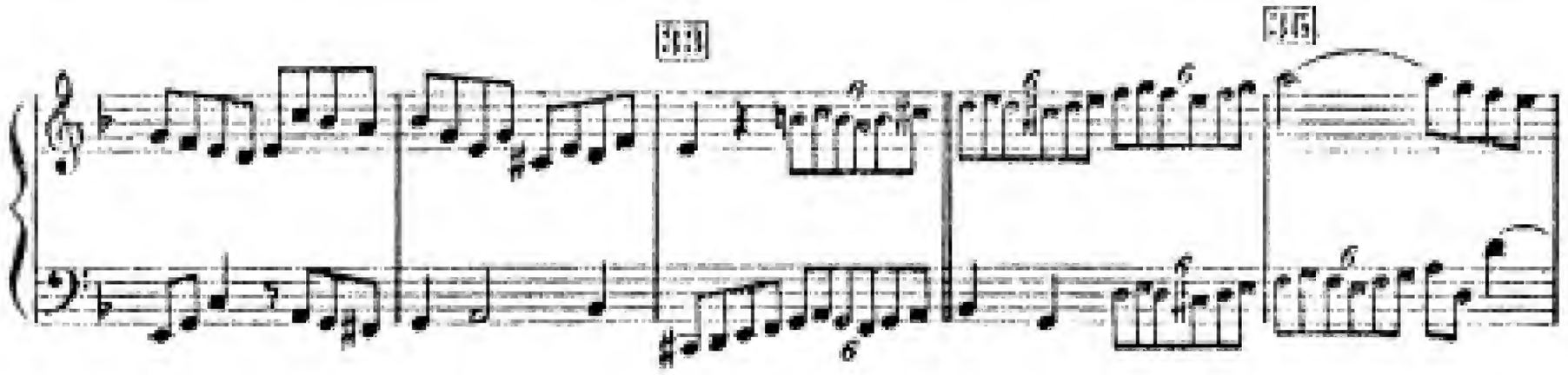
```
(T (SEMIDOWN (CAADR X))))) (CDADR X)))))
(T X))))
T4))
(SETQ CS1A (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (CADR X))
(T X))))
CS1))
(SETQ CS1B (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (CONS (ENNOTE (SEMIUP (CAADR X)))
(CDADR X)))
(T X))))
CS1))
(SETQ CS1B (TR CS1B 30))
(PROG NIL
          (SETQ REP NIL)
          (SETQ LOC (LOAD VOICE1 (QUOTE (
(HAVE SQUARE)
(TEMPO 2)
))))
          (SETQ LOC (LOAD LOC T1))
A         (SETQ LOC (LOAD LOC T2A))
          (SETQ LOC (LOAD LOC T3A))
          (SETQ LOC (LOAD LOC T4A))
          (SETQ LOC (LOAD LOC CS1A))
          (SETQ LOC (LOAD LOC CS6))
          (SETQ LOC (LOAD LOC (TR T3A 5)))
          (SETQ LOC (LOAD LOC T1))
          (SETQ LOC (LOAD LOC T2D))
          (SETQ LOC (LOAD LOC T3D))
          (SETQ LOC (LOAD LOC T4D))
          (SETQ LOC (LOAD LOC CS3))
          (SETQ LOC (LOAD LOC T1))
          (COND (REP (GO B)))
          (SETQ REP T)
          (GO A)
B         (SETQ LOC (LOAD LOC CS5))
          (RETURN LOC))
(PROG NIL
          (SETQ REP NIL)
          (SETQ LOC (LOAD VOICE2 (QUOTE (
(TEMPO 20)
(R . (1 T))
(TEMPO 2)
))))
A         (SETQ LOC (LOAD LOC (TR T1 23)))
          (SETQ LOC (LOAD LOC T2B))
          (SETQ LOC (LOAD LOC T3B))
          (SETQ LOC (LOAD LOC T4B))
          (SETQ LOC (LOAD LOC (TR T1 30)))
          (SETQ LOC (LOAD LOC T2C))
          (SETQ LOC (LOAD LOC T3C))
          (SETQ LOC (LOAD LOC T4C))
          (SETQ LOC (LOAD LOC CS1B))
          (SETQ LOC (LOAD LOC CS2))
          (SETQ LOC (LOAD LOC T3C))
          (COND (REP (GO B)))
          (SETQ REP T)
          (GO A)
B         (SETQ LOC (LOAD LOC CS4))
          (RETURN LOC))
```

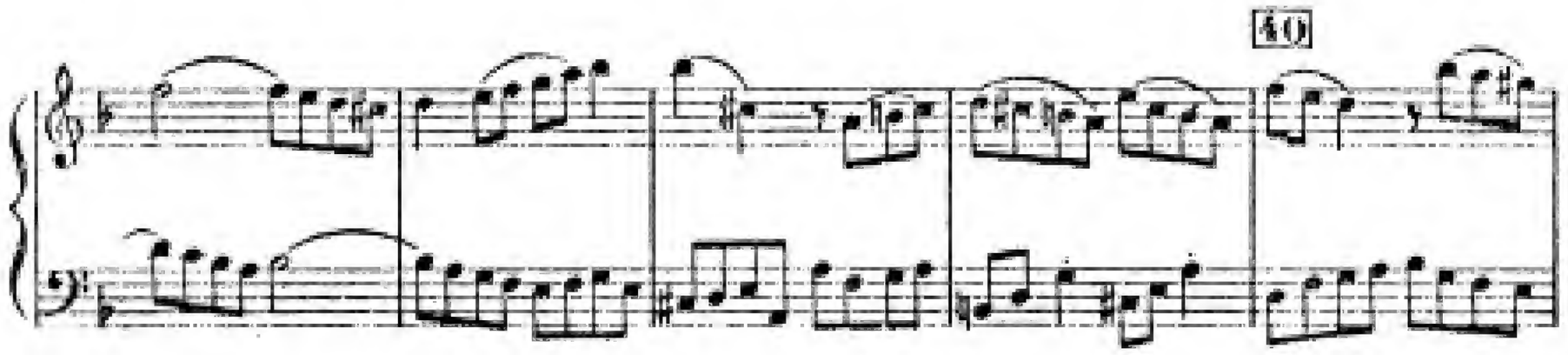
```
((NULL (ASSOC 1 X)) (CADR X))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
 (T X))))
T3))
(SETQ T3B (TR T3B 23))
(SETQ T3C (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 2 X)) (CADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (CONS (ENNOTE (SEMIJP (CAADR X)))
(CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
((EQ (CAR X) (QUOTE COM)) (CONS (CAADR X) (CDR (ASSOC 2 X))))
(T X))))
T3))
(SETQ T3C (TR T3C 30))
(SETQ T3D (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((OR (EQ (CAR X) (QUOTE PERT)) (EQ (CAR X) (QUOTE COM))) (COND
((NULL (ASSOC 3 X)) (CADR X))
((EQ (CDR (ASSOC 3 X)) (QUOTE U)) (CONS (ENNOTE (SEMIJP (COND
((NULL (ASSOC 2 X)) (CAADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (SEMIJP (CAADR X)))
(T (SEMIDOWN (CAADR X)))))) (CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (COND
((NULL (ASSOC 2 X)) (CAADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (SEMIJP (CAADR X)))
(T (SEMIDOWN (CAADR X)))))) (CDADR X))))
(T X))))
T3))
(SETQ T4A (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (CADR X))
(T X))))
T4))
(SETQ T4B (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 1 X)) (CADR X))
((EQ (CDR (ASSOC 1 X)) (QUOTE U)) (CONS (ENNOTE (SEMIJP (CAADR X)))
(CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
(T X))))
T4))
(SETQ T4B (TR T4B 23))
(SETQ T4C (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 2 X)) (CADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (CONS (ENNOTE (SEMIJP (CAADR X)))
(CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (CAADR X))) (CDADR X)))))
(T X))))
T4))
(SETQ T4C (TR T4C 30))
(SETQ T4D (MAPCAR (FUNCTION (LAMBDA (X)
(COND ((EQ (CAR X) (QUOTE PERT)) (COND
((NULL (ASSOC 3 X)) (CADR X))
((EQ (CDR (ASSOC 3 X)) (QUOTE U)) (CONS (ENNOTE (SEMIJP (COND
((NULL (ASSOC 2 X)) (CAADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (SEMIJP (CAADR X)))
(T (SEMIDOWN (CAADR X)))))) (CDADR X)))
(T (CONS (ENNOTE (SEMIDOWN (COND
((NULL (ASSOC 2 X)) (CAADR X))
((EQ (CDR (ASSOC 2 X)) (QUOTE U)) (SEMIJP (CAADR X)))
```

CANON alla Duodecima in Contrapunto alla Quinta

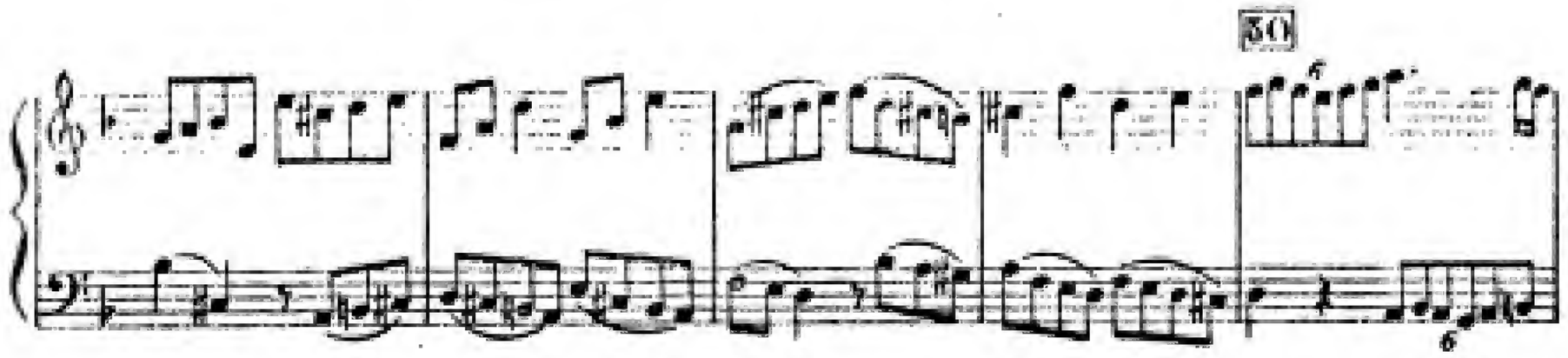








1



60

65

70

75

78

Finale